SPYWOLF

Security Audit Report

Completed on

December 10, 2023

OVERVIEW

This audit has been prepared for **Crypto Battle** to review the main aspects of the project to help investors make make an informative decision during their research process.

You will find a a summarized review of the following key points:

- ✔ Contract's source code
- ✔ Owners' wallets
- ✔ Tokenomics
- ✔ Team transparency and goals
- ✔ Website's age, code, security and UX
- ✔ Whitepaper and roadmap
- ✔ Social media & online presence

The results of this audit are purely based on the team's evaluation and does not guarantee nor reflect the projects outcome and goal

- SPYWOLF Team -

TABLE OF CONTENTS

Crypto Battle

PROJECT DESCRIPTION

According to their whitepaper:

“Crypto Battle is an investment blockchain game where you have to build the defenses of your Kingdom and fight off hordes of enemies, taking their gold for yourself. You can save gold and then exchange it for CBT (Crypto Battle Token).”

Release Date: Presale starts in December, 2023

Category: Play to earn (P2E)

CONTRACT INFO

Token Name
Crypto Battle Token

Symbol
CBT

Contract Address
0x695e8c4e49718EbF665E916e575b00330D49Ae00

Network
Binance Smart Chain

Language
Solidity

Deployment Date
Dec 07, 2023

Contract Type
Reflections token

Total Supply
10,000,000

Status
Not launched

TAXES

Buy Tax
6%

Sell Tax
6%

*Taxes can be changed in future

Our Contract Review Process

The contract review process pays special attention to the following:

- Testing the smart contracts against both common and uncommon vulnerabilities
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Blockchain security tools used:

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

TOKEN TRANSFERS STATS

Transfer Count	2
Uniq Senders	2
Uniq Receivers	2
Total Amount	16662500.261488793 CBT
Median Transfer Amount	10000000.156932509 CBT
Average Transfer Amount	8331250.130744397 CBT
First transfer date	2023-12-07
Last transfer date	2023-12-07
Days token transferred	1

SMART CONTRACT STATS

Calls Count	10
External calls	6
Internal calls	4
Transactions count	7
Uniq Callers	2
Days contract called	1
Last transaction time	2023-12-07 05:20:58 UTC
Created	2023-12-07 02:48:11 UTC
Create TX	0x52c0aa2406b1b8fdec6bb0daf5ac7c3eca0ef227ea2988df89d9259adf18a9ca
Creator	0xab90187dc2e749961dc6a896bb2f67a399886a7f

VULNERABILITY CHECK

Design Logic	Passed
Compiler warnings.	Passed
Private user data leaks	Passed
Timestamp dependence	Passed
Integer overflow and underflow	Passed
Race conditions and reentrancy. Cross-function race conditions	Passed
Possible delays in data delivery	Passed
Oracle calls	Passed
Front running	Passed
DoS with Revert	Passed
DoS with block gas limit	Passed
Methods execution permissions	Passed
Economy model	Passed
Impact of the exchange rate on the logic	Passed
Malicious Event log	Passed
Scoping and declarations	Passed
Uninitialized storage pointers	Passed
Arithmetic accuracy	Passed
Cross-function race conditions	Passed
Safe Zeppelin module	Passed
Fallback function security	Passed

THREAT LEVELS

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Low Risk

Issues on this level are minor details and warning that can remain unfixed.

Informational

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

FOUND THREATS

⚠ High Risk

Owner can change router, base pair and uniswapV2Pair addresses.
If set to inappropriate addresses, contract will halt and it will be impossible to sell.

```
function updateSwapV2Router(address newRouter) public onlyOwner {
    require(
        newRouter != address(uniswapV2Router),
        "The router already has that address"
    );
    require(newRouter != address(0), "Router cannot be the zero address");
    emit UpdateSwapV2(newRouter, address(uniswapV2Router));
    uniswapV2Router = IUniswapV2Router02(newRouter);
    router = newRouter;
    if (!_isExcluded[newRouter]) {
        _isExcluded[newRouter] = true;
    }
}

function updateBasePair(address newBasePair) public onlyOwner {
    require(
        newBasePair != address(basePair),
        "The base pair already has that address"
    );
    require(
        newBasePair != address(0),
        "Base pair cannot be the zero address"
    );

    basePair = newBasePair;
}
```

```
function updateSwapV2Pair(address newPair) public onlyOwner {
    require(
        newPair != address(uniswapV2Pair),
        "The pair already has that address"
    );
    require(newPair != address(0), "Pair cannot be the zero address");

    address _pairStatus = UniSwapFactory(uniswapV2Router.factory()).getPair(
        address(this),
        basePair
    );
    address _uniswapV2Pair = _pairStatus == zeroAddress
        ? UniSwapFactory(uniswapV2Router.factory()).createPair(
            address(this),
            basePair
        )
        : _pairStatus == newPair
        ? newPair
        : _pairStatus;

    uniswapV2Pair = _uniswapV2Pair;
    if (!_isExcluded[_uniswapV2Pair]) {
        _isExcluded[_uniswapV2Pair] = true;
    }
}
```

- Recommendation:
  - Router address and base pair address should not be changed after deployment.

FOUND THREATS

⚠ High Risk

Owner can change contract's auto swap settings.
When swapAndLiquifyEnabled is true and numTokensSellToFee is set to very low number, contract will halt and selling will fail.
When sellWithOracle is true and base pair and/or router are set to inappropriate address, contract will halt and selling will fail.

```
function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
    swapAndLiquifyEnabled = _enabled;
    emit SwapAndLiquifyEnabledUpdated(_enabled);
}

function setSwapBackSettings(uint256 _amount) external onlyOwner {
    require(_amount > 0, "Swapback amount should be at least 0");
    numTokensSellToFee = _amount;
    emit SwapAndLiquifyAmountUpdated(_amount);
}

function setSwapBackSettingsETH(uint256 _amount) external onlyOwner {
    require(_amount > 0, "Swapback amount should be at least 0");
    valueNativeSellToFee = _amount;
    emit SwapAndLiquifyAmountUpdated(_amount);
}

function setSellWithOracle(bool _status) external onlyOwner {
    require(_status != sellWithOracle, "Already set");
    sellWithOracle = _status;
}
```

- Recommendation:
  - Ensure that numTokensSellToFee is always above 1 token (consider decimals)
  - Ensure that base pair and/or router are not changed after initial deployment

FOUND THREATS

⚠ High Risk

Owner can change marketing/project/fund addresses.
If any of these addresses are set to contract that cannot receive bnb, transaction will fail and contract will halt on sell.

```
function replaceprojectWalletAddress(
    address _addr,
    address _newAddr
) public onlyOwner {
    require(_isdevWallet[_addr], "Wallet address not set previously");
    require(!_isdevWallet[_newAddr], "Wallet address already set");
    if (_isExcludedFromFee[_addr]) {
        includeInFee(_addr);
    }
    _isdevWallet[_addr] = false;
    setprojectWalletAddress(_newAddr);
}

function setprojectWalletAddress(address _addr) internal virtual {
    if (!_isExcludedFromFee[_addr]) {
        excludeFromFee(_addr);
    }
    _isdevWallet[_addr] = true;
    _projectWalletAddress = _addr;
}

function setmarketingWalletAddress(address _addr) public {
    require(
        _msgSender() == _marketingWalletAddress || _msgSender() == owner(),
        "Only marketing wallet can set this"
    );
    require(_addr != address(0), "Zero address not allowed");
    if (!_isExcludedFromFee[_addr]) {
        excludeFromFee(_addr);
    }
    _marketingWalletAddress = _addr;
}

function setfundWalletAddress(address _addr) public {
    require(
        _msgSender() == _fundWalletAddress || _msgSender() == owner(),
        "Only fund wallet can set this"
    );
    require(_addr != address(0), "Zero address not allowed");
    if (!_isExcludedFromFee[_addr]) {
        excludeFromFee(_addr);
    }
    _fundWalletAddress = _addr;
}
```

```
function _transfer(address from, address to, uint256 amount) private {
.....................
uint256 contractTokenBalance = balanceOf(address(this));

        bool overMinTokenBalance = sellWithOracle
            ? getLatestPrice(contractTokenBalance, true) >= valueNativeSellToFee
            : contractTokenBalance >= numTokensSellToFee;
        if (
            overMinTokenBalance &&
            !inSwapAndLiquify &&
            from != uniswapV2Pair &&
            swapAndLiquifyEnabled
        ) {
            //Swap to Fees
            swapAndSend(contractTokenBalance);
        }
.....................
}

function swapAndSend(uint256 contractTokenBalance) private lockTheSwap {
    // swap tokens for ETH
    swapTokensForEth(contractTokenBalance);
    // <- this breaks the ETH -> HATE swap when swap+liquify is triggered

    // capture the contract's current ETH balance.
    uint256 nativeBalance = address(this).balance;
    uint256 sendFee = nativeBalance.div(3);

    //Send to project wallets
    payable(_projectWalletAddress).transfer(sendFee);
    payable(_marketingWalletAddress).transfer(sendFee);
    payable(_fundWalletAddress).transfer(sendFee);
}
```

- Recommendation:
  - Consider using .call() or .send() without checks of the returned boolean value.

Informational

Owner can set buy/sell fees up to 5%.
Combined buy+sell = 10%.

When fees are above 0, there will be certain amount of tokens that will be deducted from every transaction that users make. Deducted amount will be as much as the fees % from total amount that user had bought, sold and/or transferred.

```
uint256 public maxTaxFee = 5;
uint256 public maxDevFee = 5;
uint256 public maxSellTaxFee = 5;

function setTaxFeePercent(uint256 taxFee) external onlyOwner {
    require(taxFee >= 0 && taxFee <= maxTaxFee, "taxFee out of range");
    _taxFee = taxFee;
    _previousTaxFee = _taxFee;
}

function setDevFeePercent(uint256 devFee) external onlyOwner {
    require(devFee >= 0 && devFee <= maxDevFee, "teamFee out of range");
    _devFee = devFee;
    _previousDevFee = _devFee;
}

function setSellTaxFeePercent(uint256 sellTaxFee) external onlyOwner {
    require(
        sellTaxFee >= 0 && sellTaxFee <= maxSellTaxFee,
        "taxFee out of range"
    );
    _sellTaxFee = sellTaxFee;
    _previousSellTaxFee = _sellTaxFee;
}
```

Informational

Owner can exclude address from fees.
When address is excluded from fees, the user will receive the whole amount of the bought, sold and/or transferred tokens.

```
function excludeFromFee(address account) public onlyOwner {
    require(!_isExcludedFromFee[account], "Account is already excluded");
    _isExcludedFromFee[account] = true;
}
```

Owner can exclude address from reflections rewards.

```
function excludeFromReward(address account) public onlyOwner {
    require(!_isExcluded[account], "Account is already excluded");
    if (_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}
```

RECOMMENDATIONS FOR

GOOD PRACTICES

1. Consider fundamental tradeoffs
2. Be attentive to blockchain properties
3. Ensure careful rollouts
4. Keep contracts simple
5. Stay up to date and track development

Crypto Battle

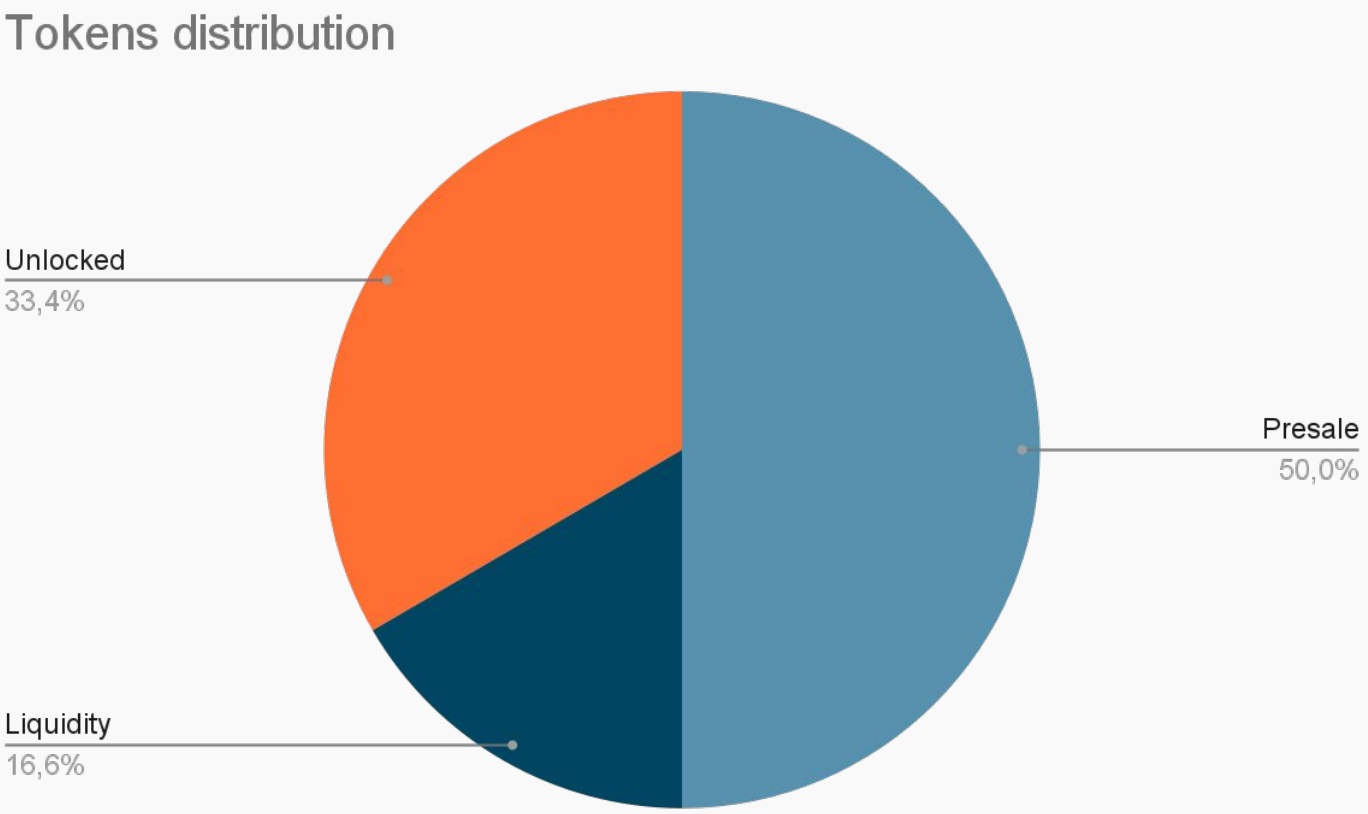
GOOD PRACTICES FOUND

- ✔ The owner cannot mint new tokens after deployment
- ✔ The owner cannot set a transaction limit

TOKENOMICS

The information about initial tokens distribution is based on Pinksale's presale page.

- 50% - Presale
- 16.6 - Liquidity
- 33.4% - Unlocked



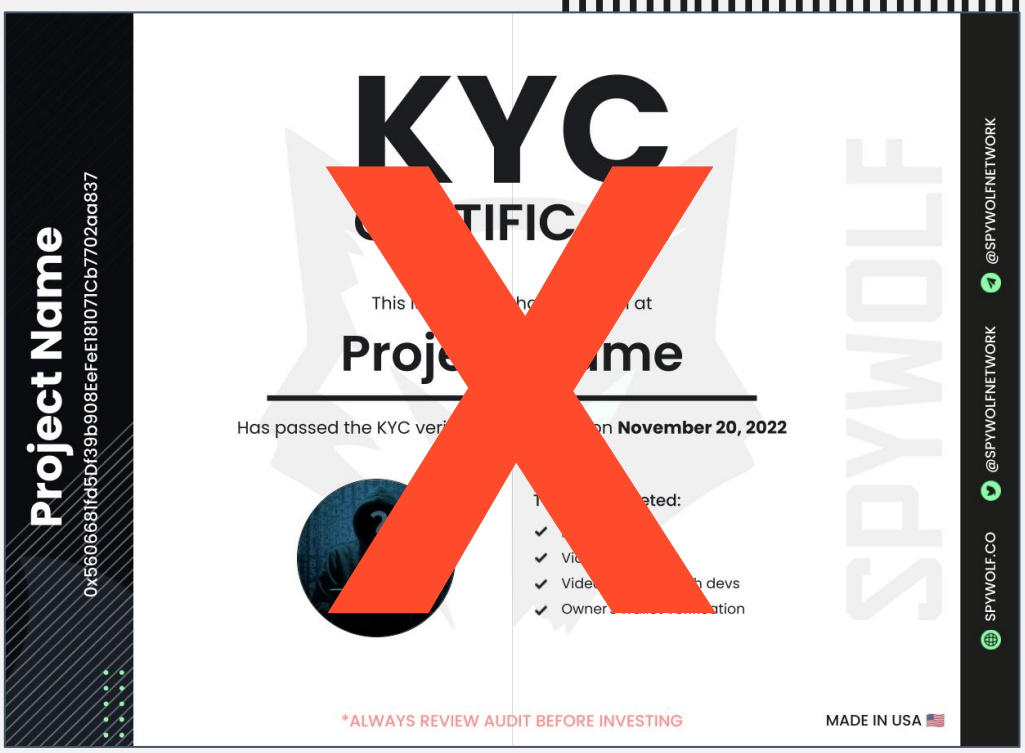
THE TEAM

⚠️ The team is annonymous

KYC INFORMATION

No KYC

We recommend the team to get a KYC in order to ensure trust and transparency within the community.



WEBSITE

Website URL
https://cryptobattlep2e.live/

Domain Registry
https://www.namecheap.com/

Domain Expiration
2024-11-02

Technical SEO Test
Passed

Security Test
Passed. SSL certificate present

Design
Single page design with appropriate color scheme and graphics.

Content
The information helps new investors understand what the product does.
No grammar mistakes found.

Whitepaper
Yes, explanatory.

Roadmap
No

Mobile-friendly?
Yes

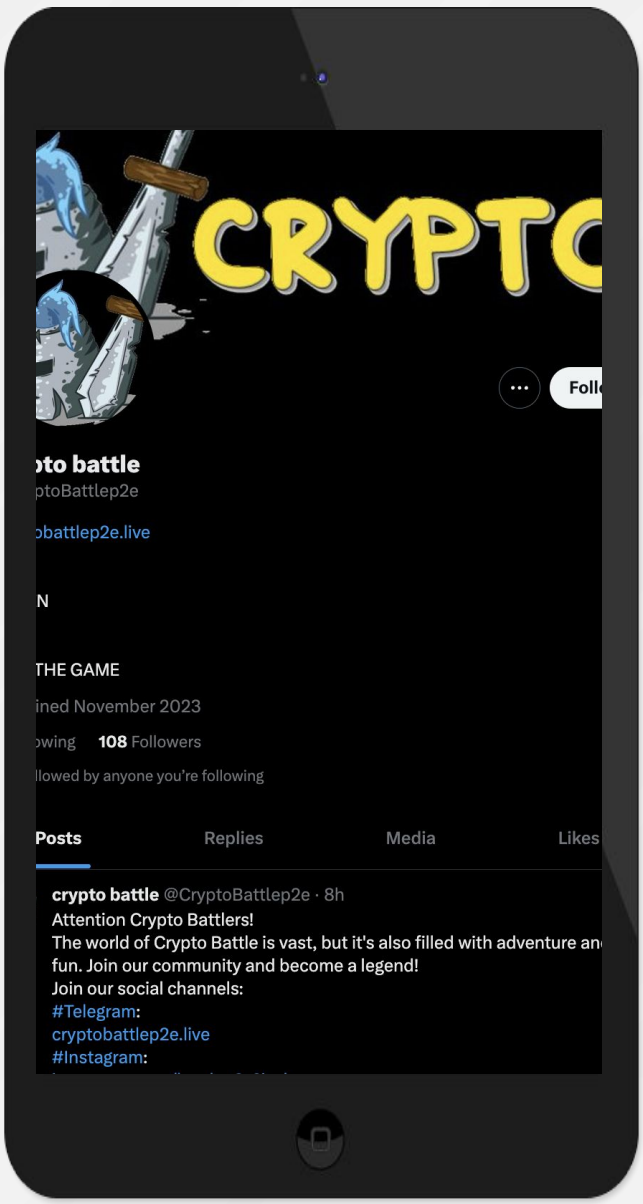
cryptobattlep2e.live

SOCIAL MEDIA

& ONLINE PRESENCE

ANALYSIS

Project's social media pages are active



Twitter

@CryptoBattlep2e

- 106 followers
- Posts frequently
- Active

Discord

- Not available

Telegram

@CRYPTOBATTL

- 768 members
- Active members
- Active mods

Medium

- Not available

SPYWOLF

CRYPTO SECURITY

Audits | KYCs | dApps
Contract Development

ABOUT US

We are a growing crypto security agency offering audits, KYCs and consulting services for some of the top names in the crypto industry.

- **OVER 700 SUCCESSFUL CLIENTS**
- **MORE THAN 1000 SCAMS EXPOSED**
- **MILLIONS SAVED IN POTENTIAL FRAUD**
- **PARTNERSHIPS WITH TOP LAUNCHPADS, INFLUENCERS AND CRYPTO PROJECTS**
- **CONSTANTLY BUILDING TOOLS TO HELP INVESTORS DO BETTER RESEARCH**

To hire us, reach out to
contact@spywolf.co or
t.me/joe_SpyWolf

FIND US ONLINE

SPYWOLF.CO

@SPYWOLFNETWORK

@SPYWOLFNETWORK

Disclaimer

This report shows findings based on our limited project analysis, following good industry practice from the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, overall social media and website presence and team transparency details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report.
While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER:
By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice.
No one shall have any right to rely on the report or its contents, and SpyWolf and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (SpyWolf) owe no duty of care towards you or any other person, nor does SpyWolf make any warranty or representation to any person on the accuracy or completeness of the report.
The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and SpyWolf hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, SpyWolf hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SpyWolf, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts, website, social media and team.
No applications were reviewed for security. No product code has been reviewed.